

JP91999-0095  
~~JAG 00-205~~  
1/8  
H. Etob, et al

Stack protection system

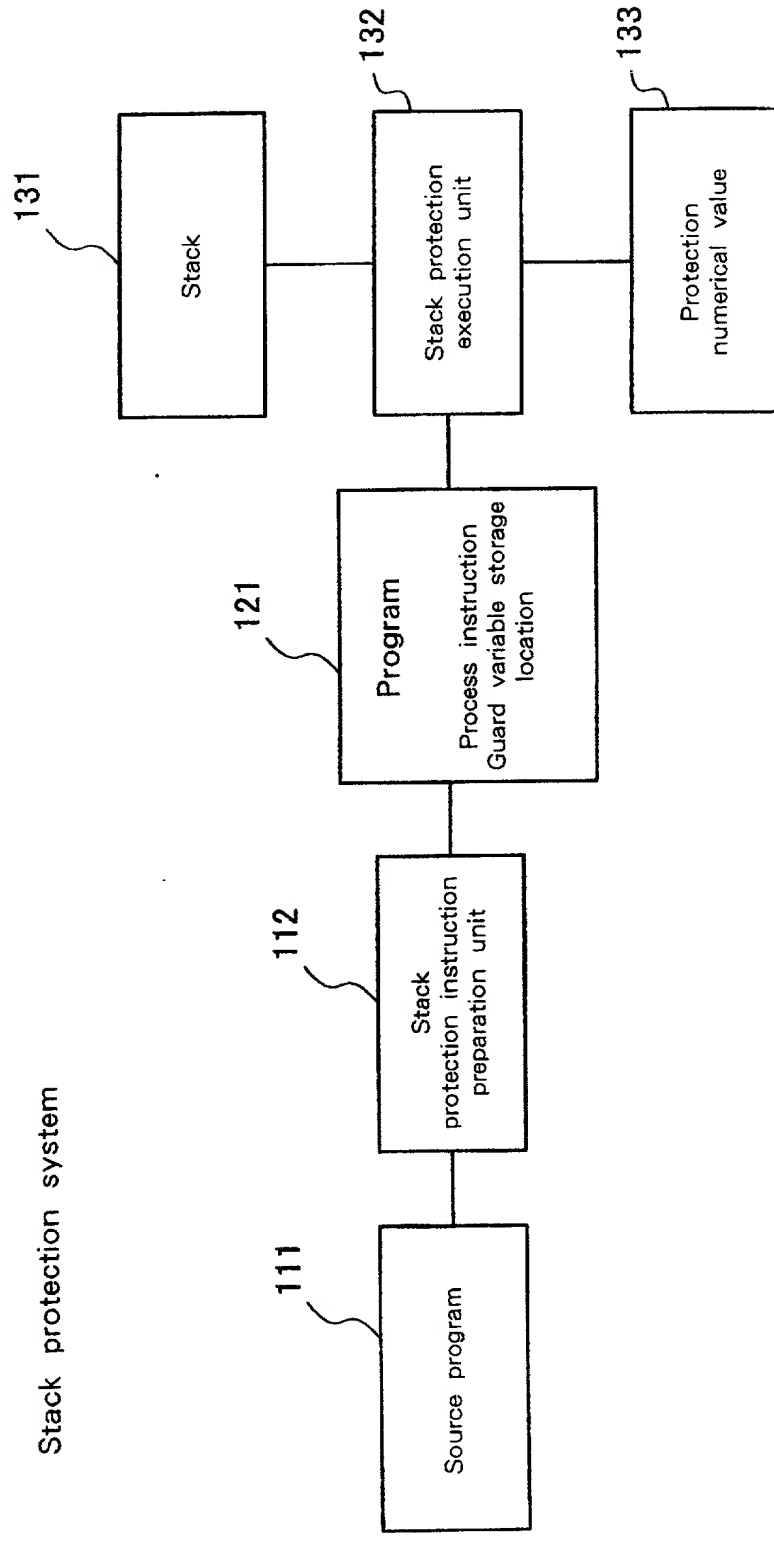


Fig. 1

Memory pattern  
when guard variable is stored

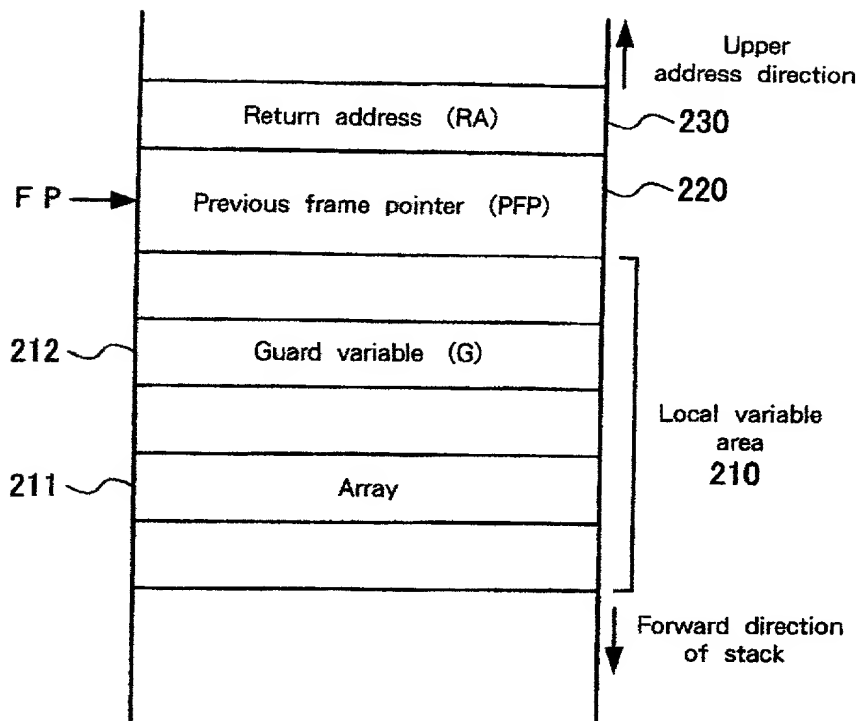


Fig. 2

Processing of stack protection  
instruction preparation unit 112

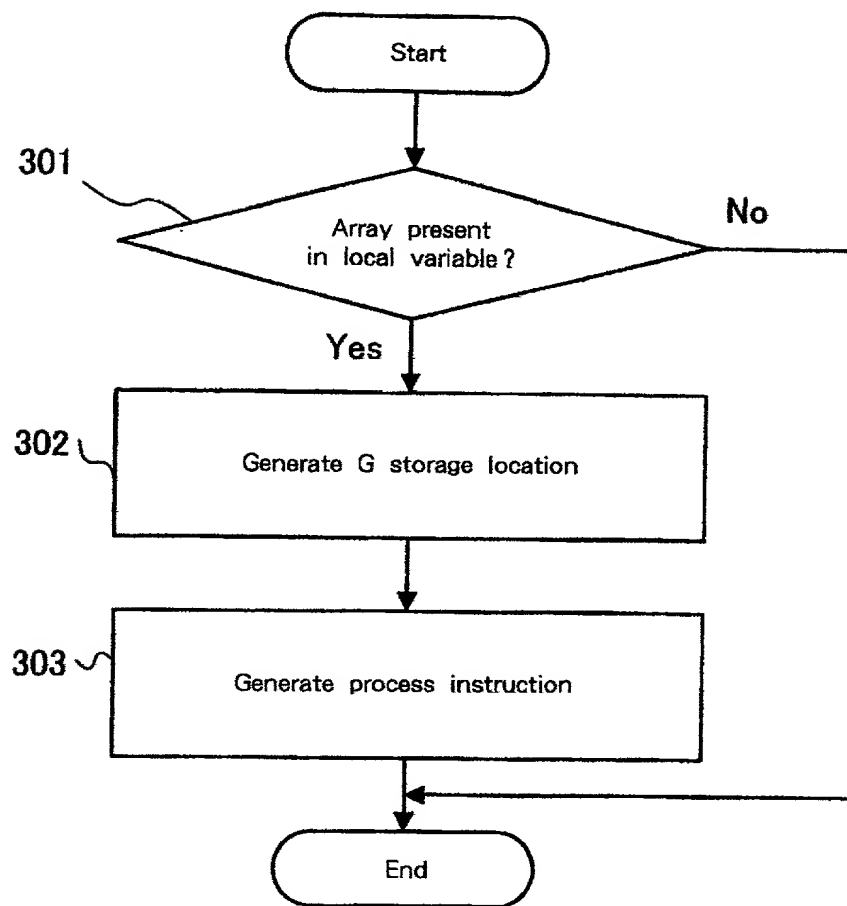


Fig. 3

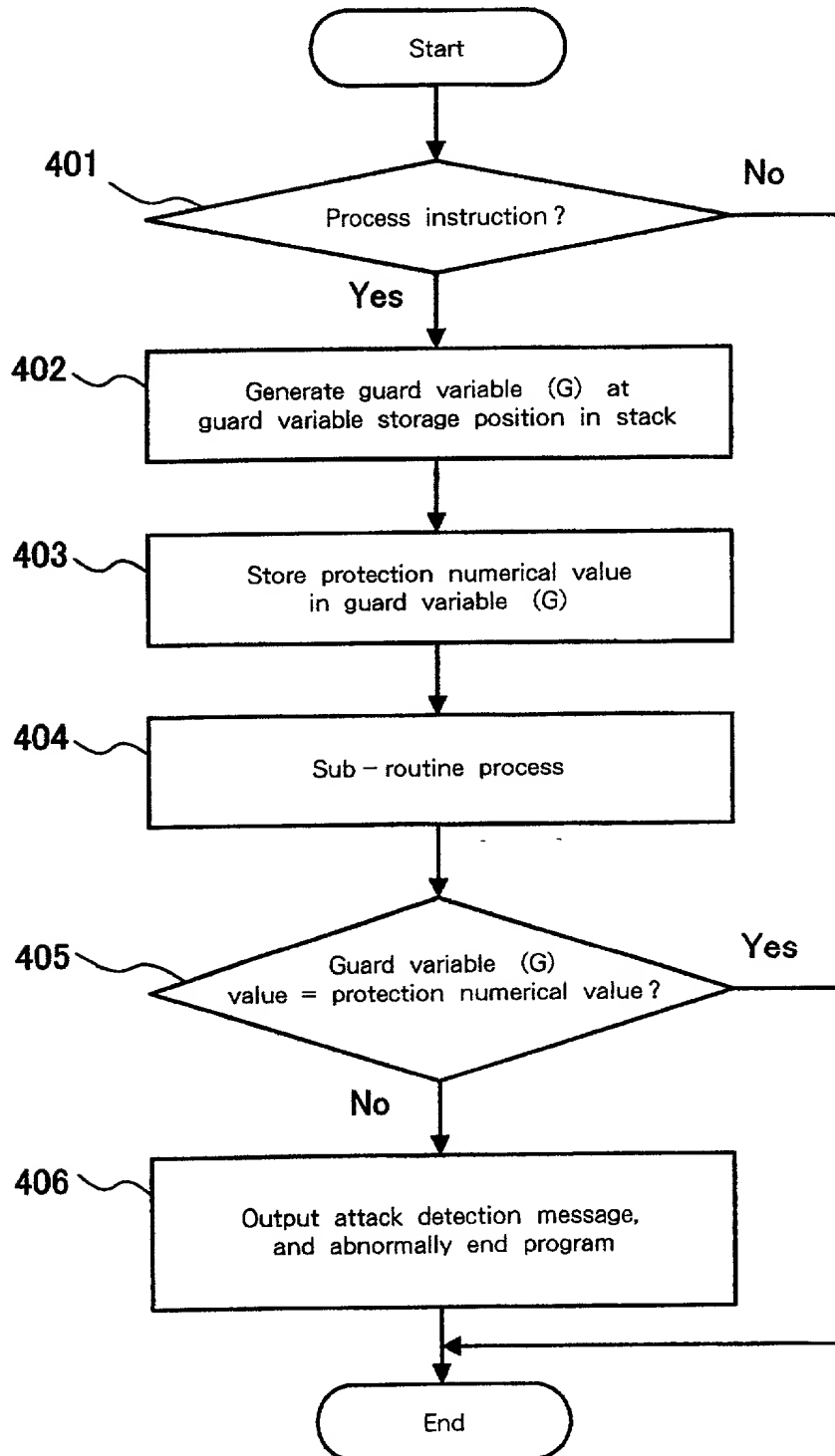
Processing of stack protection  
execution unit 132

Fig. 4

- Variable declaration  
volatile int guard;
- Function entrance  
gv = guard\_value;
- Function exit  
if (gv!= guard\_value){  
    /\*output error log \*/  
    /\*halt execution \*/  
}

Fig. 5

```

void foo()
{
    volatile int guard;    ← 601
    char buf[128];

    gv = guard_value;    ← 602
    ----
    strcpy (buf, getenv ("HOME"));
    ----
    if (gv!= guard_value){
        /*output error log */    ← 603
        /*halt execution */
    }
}

```

Fig. 6

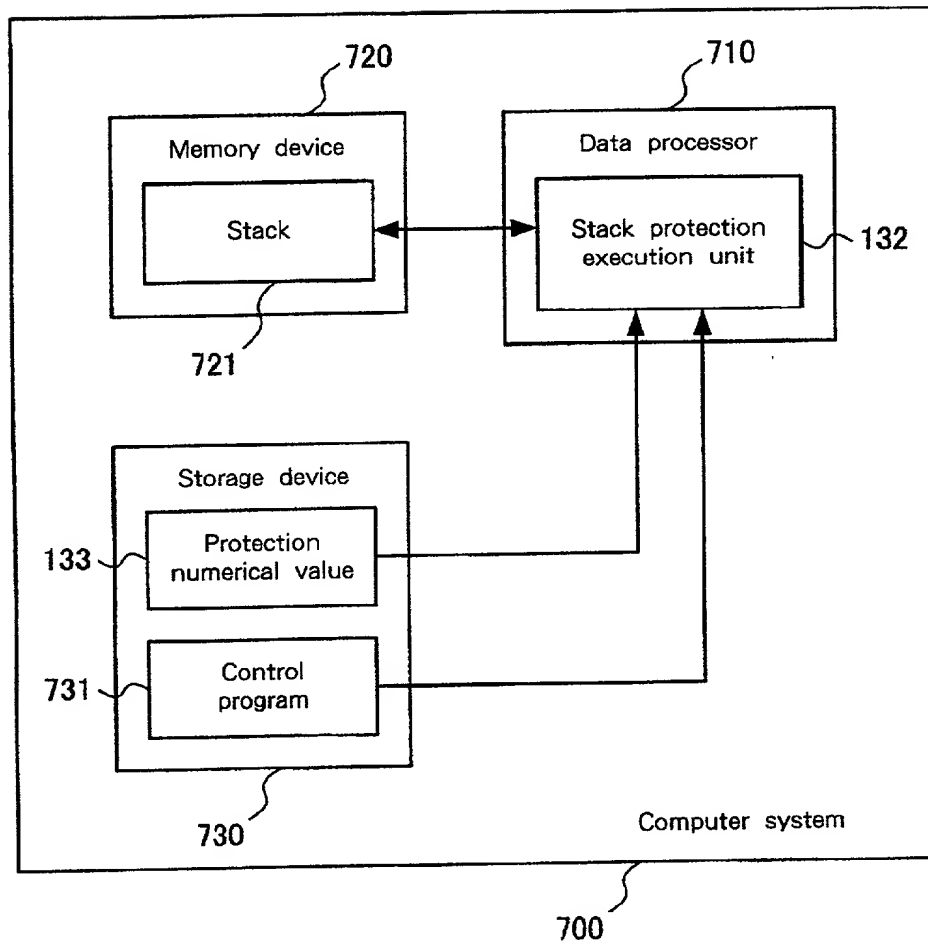


Fig. 7

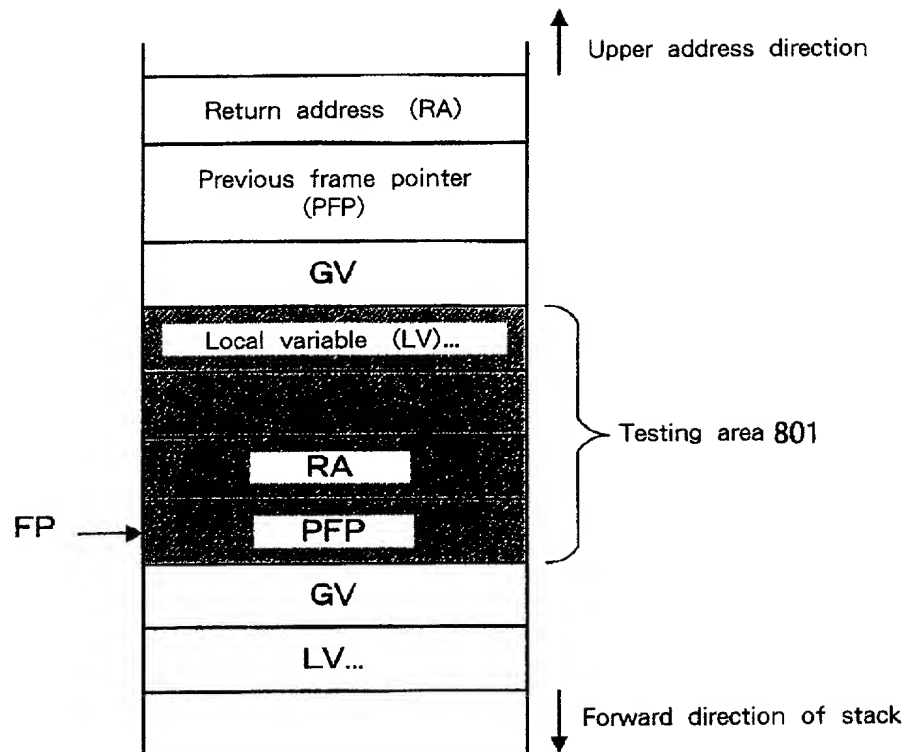


Fig. 8

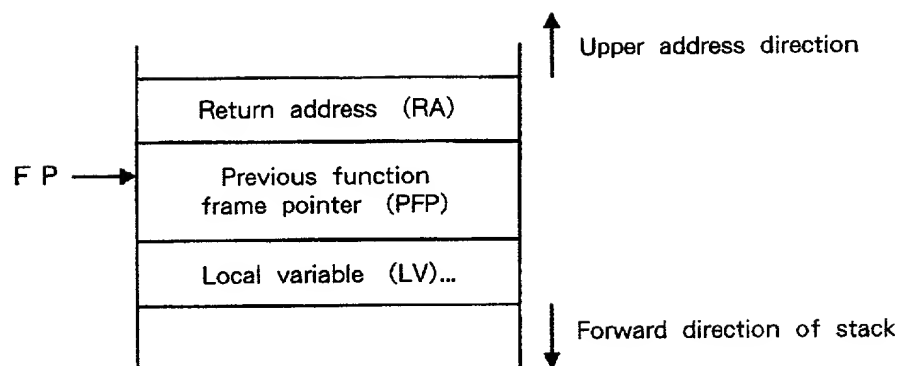


Fig. 9

```
void foo()
{
    char buf[128];
    -----
    strcpy (buf, getenv ("HOME"));
    -----
}
```

Fig. 10

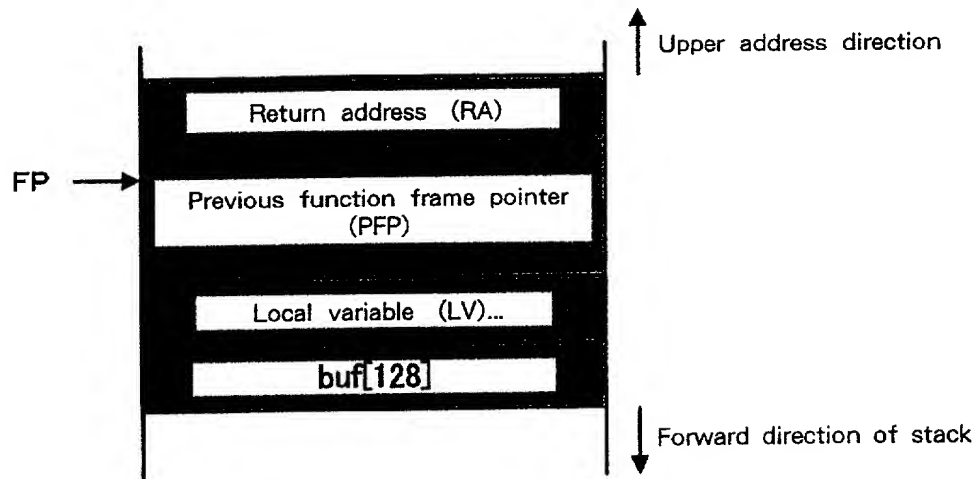


Fig. 11